

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Lovro Podgoršek

**Spletna aplikacija za organizacijo
športnih dogodkov**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Aleš Smrdel

Ljubljana 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi razvije spletno aplikacijo, ki bo omogočala organizacijo športnih dogodkov. Aplikacija naj bo zasnovana tako, da lahko športni dogodek organizira vsak prijavljen uporabnik, prav tako pa lahko uporabnik pregleduje, spreminja ali pa briše dogodke, ki jih je sam ustvaril. Poleg samega upravljanja z ustvarjenimi dogodki pa naj aplikacija posameznemu uporabniku omogoča tudi obveščanje in prijavo drugih registriranih uporabnikov. Pri organiziranju dogodka pa mora biti poskrbljeno tudi za ustrezno igrišče za posamezen šport. Zato mora aplikacija nuditi tudi vpis športnih centrov, ki vsebujejo igrišča za različne športe. Za atraktivnost in boljšo prepoznavnost naj aplikacija omogoča tudi prikaz slike športnih centrov, poleg tega pa naj tudi omogoča prikaz lokacije posameznega športnega centra. Uporabnik, ki želi organizirati športni dogodek, pa lahko izbere in rezervira ustrezna igrišča s seznama igrišč, ki so na voljo. Administrativni del aplikacije pa naj tudi še omogoča upravljanje z uporabniki in s ponudniki športnih centrov. V aplikaciji implementirajte tudi sistem za prikaz in iskanje obstoječih športnih dogodkov, ki omogoča iskanje po športu, kraju in datumu dogodka, ter sistem za obveščanje o dogodkih in prijavah ter odjavah. Pri realizaciji aplikacije uporabite tehnologije na strani odjemalca, ki bodo poskrbele za primeren prikaz strani na vseh napravah oziroma velikostih zaslonov ter tehnologije na strani strežnika, ki bodo poskrbele za hitro in odzivno delovanje spletne aplikacije.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Lovro Podgoršek, z vpisno številko 63110324, sem avtor diplomskega dela z naslovom:

Spletna aplikacija za organizacijo športnih dogodkov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Aleša Smrdela,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 4. september 2015

Podpis avtorja:

Zahvaljujem se mentorju doc. dr. Alešu Smrdelu za pomoč pri izdelavi diplomskega dela. Prav tako se zahvaljujem svoji družini za vso podporo tekom celotnega študija. Zahvaljujem se tudi Alji Randl, Jaki Šušteršiču in vsem svojim študijskim prijateljem, ki so me spodbujali in mi stali ob strani.

Družini in vsem, ki so me podpirali in
spodbujali pri študiju.

Kazalo

Povzetek

Abstract

| | | |
|----------|--------------------------------------------------------|-----------|
| 1 | Uvod | 1 |
| 2 | Uporabljeni jeziki in orodja | 3 |
| 2.1 | Jeziki | 3 |
| 2.2 | Orodja | 6 |
| 3 | Ogrodje Ruby on Rails, knjižnice in arhitektura | 9 |
| 3.1 | Ruby on Rails | 9 |
| 3.2 | Knjižnice | 10 |
| 3.3 | Arhitektura | 10 |
| 4 | Analiza in načrtovanje | 13 |
| 4.1 | Uporabniške vloge | 13 |
| 4.2 | Načrtovanje podatkovne baze | 15 |
| 5 | Implementacija | 17 |
| 5.1 | Postavitev ogrodja in konfiguracija | 17 |
| 5.2 | Opis uporabniškega vmesnika | 18 |
| 5.3 | Funkcionalnosti | 31 |
| 5.4 | Postavitev strežnika in zagon aplikacije | 37 |

KAZALO

| | | |
|----------|---------------------------|-----------|
| 6 | Sklepne ugotovitve | 39 |
| 6.1 | Nadaljnje delo | 40 |

Seznam uporabljenih kratic

| kratica | angleško | slovensko |
|-------------|------------------------------------------------|------------------------------------------------------|
| HTML | Hyper Text Markup Language | Jezik za označevanje besedila |
| CSS | Cascading Style Sheets | Kaskade slogovne podloge |
| AJAX | Asynchronous JavaScript and XML | Asinhroni JavaScript in XML |
| JSON | JavaScript Object Notation | Notacija objektov JavaScript |
| URL | Uniform Resource Locator | Enolični krajevnik vira |
| MVC | Model-View-Controller | Model-Pogled-Kontroler |
| ORM | Object-relational mapping | Objektno-relacijsko preslikovanje |
| HTTP | Hyper Text Transfer Protocol | Protokol za prenos hiperteksta |
| CSV | Comma-separated values | Z vejico ločene vrednosti |
| API | Application programming interface | Vmesnik za dostop do podatkov |
| SVG | Scalable Vector Graphics | Stopnjevane vektorske slike |
| XML | Extensible Markup Language | Razširljiv označevalni jezik |
| XUL | XML User Interface Language | XML jezik za grafični vmesnik |
| DOM | Document Object Model | Objektni model dokumenta |
| SQL | Structured Query Language | Strukturiran povpraševalni jezik |
| XSLT | Extensible Stylesheet Language Transformations | Transformacija jezika razširljivih slogovnih polj |

Povzetek

Naše diplomsko delo prikazuje razvoj spletne aplikacije za organizacijo športnih dogodkov, ki je napisano s pomočjo ogrodja Ruby on Rails. Glavni cilj aplikacije je omogočiti posameznikom, da organizirajo ali sodelujejo pri dogodkih v njihovi okolici.

Prvo poglavje vsebuje kratek povzetek vseh orodij in programskih jezikov, ki smo jih uporabili tekom izdelave diplomske naloge, kot so HTML, AJAX, JavaScript, Ruby in Ruby on Rails.

V drugem poglavju smo predstavili načrtovanje naše podatkovne baze in analizo zahtev za implementacijo. V naslednjem poglavju predstavimo vse tri uporabniške vloge: administrator, lastnik športnega centra in navaden uporabnik. Prav tako smo predstavili funkcionalnosti, na katerih temeljijo uporabniške vloge. Aplikacijo smo za lažjo predstavitev bralcu postavili tudi na lastni strežnik.

Ključne besede: Ruby on Rails, spletna aplikacija, športni dogodek.

Abstract

Our thesis demonstrates an example implementation of a web application, used for organizing sports events, written using the Ruby on Rails framework. The application's main goal is to enable individuals to organize or take part in an event in their local area.

The first chapter includes a brief summary of all the tools and programming languages we used in our thesis such as HTML, AJAX, JavaScript, JSON, Ruby and Ruby on Rails.

In the second chapter, we present an overview of our database schema design and the implementation of our application's backend and user interface. The following chapter introduces all three major implemented user roles: administrator, sports society manager and regular user. We also present the user paths of our application's most important functionality more thoroughly. The application was also deployed to a private server, to better demonstrate real world usability.

Keywords: Ruby on Rails, web application, sport event.

Poglavje 1

Uvod

Današnja družba je preveč pod stresom in ima ogromno obveznosti. Zato se tudi vedno manj ljudi zaveda, da se je kljub vsakodnevnim obveznostim potrebno ukvarjati s športom. Ravno zaradi časovne omejitve pa težko najdemo ljudi, ki bi se želeli rekreirati ter družiti pri športni aktivnosti.

Popoldanska in večerna rekreacija na športnih igriščih in v športnih centrih je zelo razširjena, vendar kot posamezniki kar težko samostojno pristopimo. Večina ljudi, ki prihaja na rekreacijo, že ima obstoječe ekipe. Občasno se zgodi, da ekipam manjkajo člani in so posamezniki brez ekipe dobrodošli, vendar ni vedno tako. Zato se je posamezniku težko ukvarjati z rekreacijo, če nima zadosti prijateljev. Tudi ekipe, ki redno obiskujejo rekreacijo in imajo rezervirano dvorano ali igrišče (odvisno od športa), imajo velikokrat težavo z organizacijo rekreacije in iskanjem članov za tvorjenje ekip. Za zbiranje in informiranje članov uporabljajo mobilne telefone. To početje je dokaj drago ter zamudno. Vsakega člana posebej je potrebno kontaktirati preko telefonskega klica ali mu poslati nekaj tekstovnih sporočil, kar se na koncu meseca pozna na računu za telefon.

Ni problem samo z ekipnimi športi, tudi s kolesarstvom in pohodništvom se je dosti prijetneje ukvarjati v družbi kot individualno. Prednosti so druženje, spoznavanje novih ljudi, zabava... Poleg druženja je prednost skupinskega pohodništva tudi varnost, saj imaš ob morebitni nezgodi ob sebi ljudi, ki

ti lahko v trenutku priskočijo na pomoč. Kot individualist bi bil v takem primeru prepuščen samemu sebi.

Rešitev, ki bi bila ugodna tako za posameznika kot tudi za ekipe, je razvoj spletne aplikacije, ki bi omogočala organizacijo športnega dogodka ter prijavo posameznika na določeno športno aktivnost. Ravno ta problem bomo podrobneje obravnavali v tej diplomski nalogi. Razvili bomo spletno aplikacijo, ki bi vsakomur, ki bi se želel ukvarjati s športom, omogočala enostavnejšo priključitev k športnim aktivnosim, pri tem pa še zabavo in druženje z ljudmi.

V prvem delu diplomskega dela se bomo dotaknili osnovnih tehnologij, HTML-ja, CSS-ja ter JavaScript-a, s katerimi bomo razvili spletno aplikacijo. Poleg vseh osnovnih tehnologij, ki so nujno potrebne za prikaz spletne strani, bomo uporabili ogrodje za razvoj aplikacij na strani strežnika Ruby on Rails. V drugem delu pa se bomo ustavili pri predstavitvi aplikacije in njene implementacije ter predstavitvi funkcionalnosti, ki jih aplikacija omogoča. V zadnjem poglavju bomo analizirali prednosti, ki smo jih dosegli, poleg tega pa bomo opisali še možnosti za nadaljnji razvoj.

Poglavje 2

Uporabljeni jeziki in orodja

V tem poglavju si bomo malo podrobneje pogledali, katere jezike za opis spletnih strani, orodja in tehnologije smo uporabili za razvoj. Jezika na strani odjemalca sta HTML in CSS, jezik na strani strežnika pa je Ruby. Orodja, s katerimi smo razvijali aplikacijo, so RubyMine, MySQL Workbench, phpMyAdmin ter Googlov Development Tools. Za reviziranje programske kode smo uporabili GIT, za pošiljanje obvestil pa aplikacijo Mandrill (MailChimp). Za sprotno testiranje in ugotavljanje napak pa smo uporabili sodobne brskalnike - Google Chrome, Mozilla Firefox in Internet Explorer.

2.1 Jeziki

V tem podpoglavju si bomo natančneje ogledali jezike, s katerimi smo se srečali pri implementaciji naše spletne aplikacije. Osnovni jezik za prikaz statične spletne strani je HTML, CSS pa poskrbi za njeno obliko. S pomočjo Ruby-ja in AJAX-a postane spletna aplikacija dinamična. Uporabljata se za asinhrono komunikacijo med odjemalcem in strežnikom ter za prenos podatkov med njima.

2.1.1 HTML in CSS

HTML [7] je standardni označevalni jezik za izdelavo spletnih strani. S pomočjo HTML jezika določimo strukturo in semantiko dokumenta. Spletni brskalniki (Google Chrome, Mozilla Firefox, Internet Explorer, ...) so orodja, ki preberejo HTML dokument in prikažejo njegovo vizualno vsebino. Začetki razvoja segajo v leto 1980, ko je fizik Tim Berners Lee v CERN-u predlagal prototip sistema za pošiljanje in deljenje dokumentov. Za uspešno delovanje je avtor razvil še prototip ENQUIRE, ki je predhodnik sedanjega svetovnega spleta (*World Wide Web*). Prva verzija HTML-ja je bila javnosti predstavljena leta 1990, imenovana HTML 1.0. Definiranih je bilo 18 preprostih značk za prikaz osnovne spletne strani. Nekatere izmed njih so bile: definicija naslova (<TITLE>), povezava (<A>), odstavek (<P>), seznam (<LISTING>) in druge [8]. Enajst osnovnih značk, ki jih vsebuje tudi verzija HTML 4.0, je v uporabi še danes.

Zadnja večja sprememba je standardizacija HTML5 [9], ki je bila oktobra 2014. Prinaša novosti predvsem pri področju večpredstavitvenih vsebin, dvo-dimenzionalni grafiki ter pri tipih vnosnih polj pri obrazcih. Pojavijo se nove značke, ki še enostavneje in natančneje opisujejo strukturo HTML datoteke.

CSS [3] je jezik za oblikovanje spletne strani. Uporablja se za oblikovanje spletnih strani in uporabniških vmesnikov, možno ga je tudi vključiti v XML, SVG in XUL datoteke. Omogoča spreminjanje barve, oblike in velikosti pisave, razmike, dodajanje obrob in drugih elementov oblikovanja.

Zaradi slabega izgleda, ki je nastal ob uporabi HTML-ja, je Håkon Wium Lie leta 1994 organizaciji W3C poslal predlog o standardizaciji CSS-ja. Predlog je bil sprejet in konec leta 1996 je CSS 1 postal standardiziran. Pri sedanjem razvoju spletnih strani je uporaba CSS pravil samoumevna.

2.1.2 JavaScript in AJAX

JavaScript [10] je dinamični programski jezik, s katerim postane osnovna HTML stran odzivna na uporabniške zahteve. Njegov interpreter se upora-

blja kot del brskalnikov, ki odjemalski strani (*client-side*) omogoča interakcijo z uporabnikom, nadzoruje brskalnik, asinhrono komunicira s strežnikom ter spreminja vsebino prikazane strani. Razvit je bil neodvisno od Jave, čeprav sta si pri nekaterih podrobnostih zelo podobna.

Razvijalci ga uporabljajo, kadar naloge niso mogoče samo s statičnim HTML-jem. Uporabljajo ga za prikaz skrite vsebine, asinhrono komuniciranje s strežnikom, odpiranje oken, preverjanje pravilnosti in podobno. Na žalost različni brskalniki različno podpirajo JavaScript objekte. Za podporo vsem brskalnikom je potrebno napisati več različnih funkcij.

AJAX [1] je skupina med seboj povezanih spletnih razvojnih tehnik, uporabljenih na strani odjemalca, za izdelavo asinhronih spletnih aplikacij. Aplikacija lahko s pomočjo AJAX-a in XMLHttpRequest zahtevka (*XMLHttpRequest*) asinhrono komunicira s strežnikom. Omogoča spremembo dela strani, pri tem pa ni potrebe po ponovnem nalaganju celotne strani. Jesse James Garrett je v članku [24] leta 2005 pojasnil, da AJAX vključuje:

- HTML in CSS za osnovni prikaz strani,
- DOM za dinamičnost strani,
- XML in XSLT za interakcijo in manipulacijo podatkov,
- XMLHttpRequest za asinhrono pridobivanje podatkov,
- JavaScript, ki povezuje vse tehnologije v celoto.

2.1.3 Ruby

Ruby [20] je dinamičen, odprtokodni, objektno usmerjen programski jezik. Namenjen je dinamični gradnji vsebin na strežniški strani. Njegov fokus je usmerjen v enostavnost in učinkovitost. Z enostavnimi ukazi lahko naredimo nekaj, kar je v drugih programskih jezikih lahko zelo kompleksno. Po mnenju avtorja Yukihiro Matsumota naj bi Ruby vplival na razvoj drugih programskih jezikov, kot so: Perl, Smalltalk, Eiffel, Ada in Lisp.

Ideja je bila ustvariti takšen programski jezik, ki je učinkovit in zabaven, pri tem pa slediti principom dobrega načrtovanja uporabniških vmesnikov. Dejal je, da se programerji preveč osredotočajo na računalnike (stroje), da bi programi delovali čim hitreje, da so učinkoviti in še mnogo drugega. V resnici se je potrebno osredotočiti na ljudi. Ljudje smo šefi (mojstri), stroj (suženj) je samo orodje, s katerim operiramo [21].

2.2 Orodja

2.2.1 RubyMine

RubyMine [22] je produkt podjetja JetBrains. Predstavljen je kot najbolj pameten urejevalnik Ruby kode. Zaradi vgrajenega dopolnjevalnika kode, hitre navigacije po kodi, lahkega odpravljanja hroščev (*Software bugs*), enostavne uporabe konzole ter podpore različnim jezikom je nadvse primeren za resen razvoj. Ima vgrajeno orodje za delo s sistemi za verzioniranje kode. Reviziranje na prejšno verzijo nam prihrani ogromno časa. Pozna veliko jezikov, ki se uporabljajo za razvoj spletnih aplikacij, in njihovo sintakso obarva ter analizira že med samim pisanjem.

Zaradi vseh teh funkcionalnosti, ki jih omogoča, je razvoj potekal zelo hitro. Program opozori na napako takoj, ko zaključimo s pisanjem stavka. Na ta način nas obvaruje pred izgubo dragocenega časa, ki bi ga porabili pri iskanju napak. Cena licence za komercialne uporabnike se giblje okoli 200€, vendar je za študente in učitelje Univerze v Ljubljani brezplačna.

2.2.2 MySQL Workbench

MySQL Workbench [15] je orodje za vizualizacijo podatkovnih baz. Primeren je za načrtovalce, razvijalce in administratorje podatkovnih baz. S pomočjo grafičnega vmesnika lahko gradimo, modeliramo in načrtujemo podatkovne baze. Orodje nam avtomatično pretvori naš podatkovni model v MySQL podatkovno bazo. Razvijalci lahko izvajajo SQL poizvedbe in jih

analizirajo. Administratorji pa lahko spremljajo delovanje podatkovne baze preko grafičnega vmesnika. Omogoča tudi migracijo podatkov iz drugih baz na MySQL podatkovno bazo. Orodje je na spletu prosto dostopno vsem uporabnikom. [16].

2.2.3 phpMyAdmin

PhpMyAdmin [19] je brezplačno in odprtokodno orodje, napisano v programskem jeziku PHP. Namenjeno je upravljanju z MySQL podatkovnimi bazami preko spletnega brskalnika. Omogoča tudi kreiranje, spreminjanje in brisanje podatkovnih baz, tabel, polj ali vrstic. Prav tako omogoča izvajanje SQL stavkov ter upravljanje s pravicami uporabnikov. Omogoča pa tudi uvoz (*import*) podatkov v obliki CSV ali SQL datotek ter izvoz (*export*) podatkov v formate CSV, SQL, XML, PDF, Microsoft Word, Microsoft Excel in druge.

2.2.4 GitHub

GitHub [5] je storitev, ki omogoča uporabo GIT [4] storitve na spletu. Ponuja porazdeljeno kontrolo revizije programske kode z izvorno kodo. Za razliko od GIT-a je GitHub spletna storitev z grafičnim vmesnikom. Poleg tega zagotavlja tudi nadzor dostopov, sledenje hroščem (*bug tracking*), upravljanje opravil (*task management*) in dokumentacijo za vsak projekt.

GitHub ponuja plačljivi zasebni arhiv za lastne projekte ter brezplačen javni arhiv, ki se običajno uporablja za gostovanje odprtokodnih projektov. Leta 2015 ima že preko 9 milijonov uporabnikov ter 21,1 milijonov arhivov, zaradi česar je največji gostitelj izvirne kode na svetu.

Poglavje 3

Ogrodje Ruby on Rails, knjižnice in arhitektura

3.1 Ruby on Rails

Ruby on Rails (RoR) ali preprosto Rails je odprtokodno ogrodje (*framework*), ki deluje po principu MVC (model-pogled-kontroler) in je primeren za razvoj spletnih aplikacij z uporabo strežniškega jezika Ruby. Za prenos podatkov uporablja standardizirane formate JSON ali XML. Rails poleg MVC omogoča tudi uporabo drugih razvojnih (*Software Engineering*) vzorcev, kot so: programiranje brez ponavljanja obstoječe kode (*DRY - don't repeat yourself*) ter programiranje "manj je več" [23].

Odprtokodno ogrodje Rails je razvil David Heinemeier Hansson leta 2004 v okviru projekta Basecamp v istoimenskem podjetju. Na začetku ni bilo pretiranega zanimanja zanj. Večji premik se je zgodil leta 2006, ko je Apple najavil, da bo dodal podporo v Mac OS X Leopard, ki je izšel oktobra 2007 [13].

Sedaj je v uporabi verzija 4.2, ki je izšla 14. decembra 2014. Zato smo tudi našo spletno aplikacijo razvijali v tej verziji ogrodja. Trenutno je v uporabi verzija jezika Ruby 2.3, zaradi stabilnosti pa smo uporabili verzijo 1.9.3, ki je izšla 31. oktobra 2011.

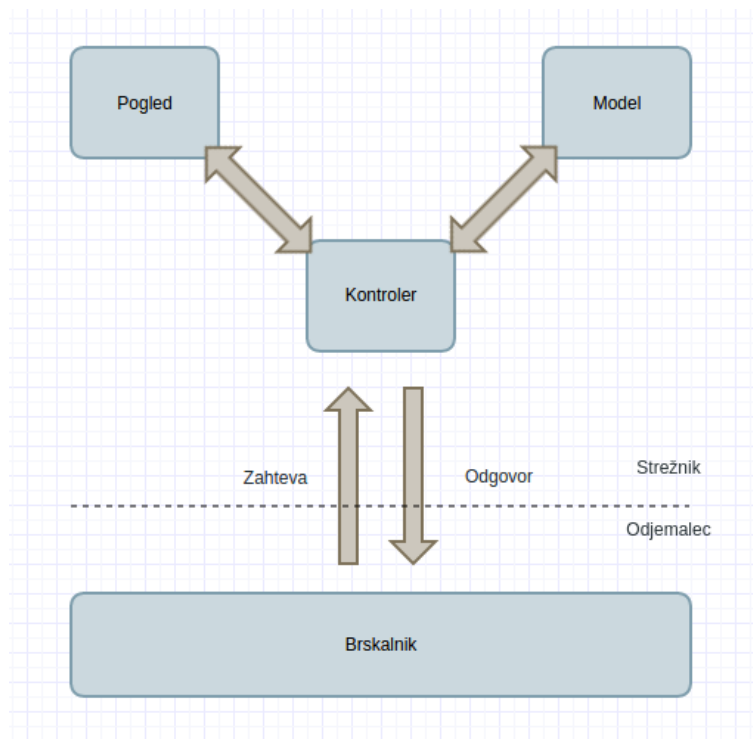
3.2 Knjižnice

- **jQuery** [11] - je JavaScript knjižnica, ki deluje na različnih operacijskih sistemih (*cross-platform*). Vsebuje funkcije, ki se pogosto uporabljajo pri kreiranju nove spletne strani. To so: odzivanje na dogodke, odpiranje novih oken, spreminjanje elementov, prikaz animacij in klic API servisov preko AJAX-a. S pomočjo jQuery-ja je JavaScript enostavneje uporabljati.
- **jQueryUI** [12] - je zbrana množica grafičnih elementov za enostavnejšo gradnjo uporabniških vmesnikov. Temelji na knjižnici jQuery in je napisana v JavaScript-u. Knjižnica je odlična izbira tako za visoko interaktivne aplikacije kot tudi za enostavne vmesnike, kjer je potreben le gradnik (*widget*) za datum ali čas.
- **Bootstrap** [2] - je brezplačno in odprtokodno ogrodje za gradnjo spletnih strani in aplikacij. V osnovi je zgrajeno iz HTML-ja in CSS-a, za dinamične akcije pa uporablja JavaScript. Njegova uporaba se je razširila v zadnjih letih predvsem zaradi trenda uporabe pametnih telefonov, s tem pa se je pojavila potreba po prilagajanju spletnih strani za manjše zaslone.

3.3 Arhitektura

Pri implementaciji naše spletne aplikacije smo se odločili za arhitekturo MVC [14], ki je prikazana na sliki 3.1. Funkcije iz različnih področij so lepo ločene med seboj, kar omogoča preprost in enostaven pogled v kodo. Kot že ime pove, je celotna arhitektura ločena na tri komponente:

- Model (ang. Model) je element, ki aplikacijo povezuje z virom podatkov, kot je podatkovna baza, spletna storitev preko JSON-a ali kaj drugega. Vsebuje logiko za nadzor in obdelavo podatkov. V njem pa



Slika 3.1: Arhitektura MVC v ogrodju Ruby on Rails.

se tudi izvaja preverjanje pravilnosti podatkov, preden se pošljejo v podatkovno bazo.

- Pogled (ang. View) je element, ki služi za izris uporabniškega vmesnika. V njem se uporabljajo podatki, ki jih je posredoval kontroler. Rezultat na izhodu je statična HTML stran, lahko pa tudi dinamično generiran podatkovni vir, kot je JSON ali XML.
- Kontroler (ang. Controller) je element, v katerem se nahaja vsa poslovna logika. Je nekakšen most med brskalnikom, pogledom in modelom. Nadzira izvajanje prejetega HTTP zahtevka ter poskrbi za ustrezen odgovor.

Poglavje 4

Analiza in načrtovanje

Ideja aplikacije je zagotoviti športno dejavnost vsakomur, ki si to želi. Dogodek je obravnavan kot predmet, ki ga je mogoče ustvariti, se nanj prijaviti, spremljati prijave ter ga urejati in izbrisati.

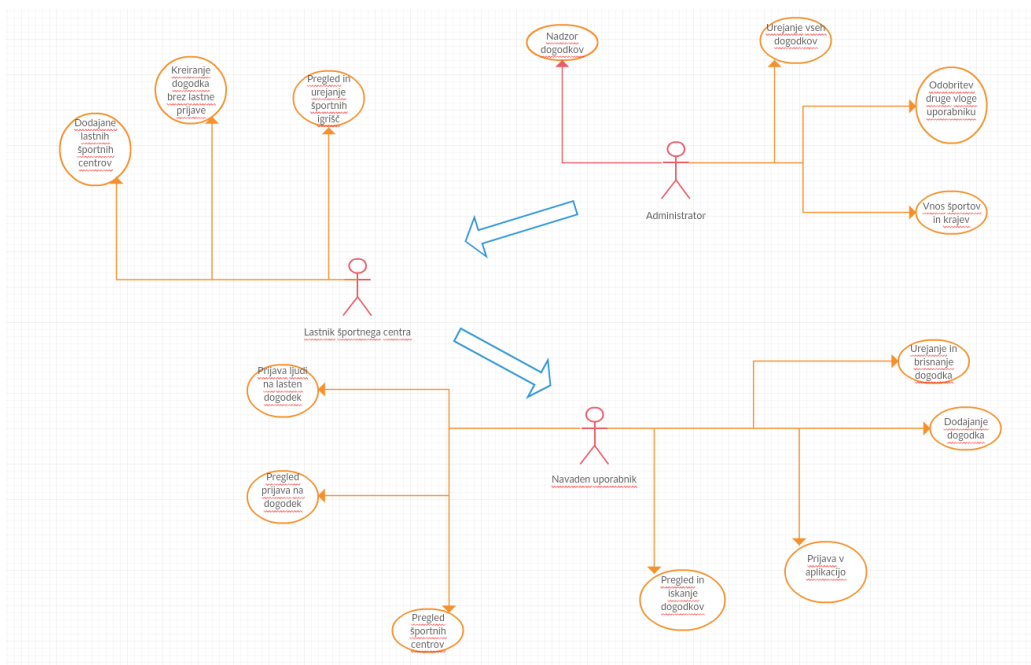
Pri razvoju spletne aplikacije smo sledili trendom iterativnega razvoja programske opreme. To pomeni, da smo se najprej osredotočili na kritične točke naše aplikacije (enostavna prijava in odjava od dogodka). Nato smo se vračali nazaj, da smo odpravljali napake, izboljševali obstoječe funkcionalnosti in dodajali takšne, ki so se nam med razvojem zdele smiselne.

Z analizo smo zbrali funkcionalnosti, ki naj bi jih naša aplikacija vsebovala. Te funkcionalnosti pa smo s pomočjo diagarama razdelili na več uporabniških vlog.

4.1 Uporabniške vloge

Kakor se za vsak dober razvoj spodobi, smo najprej naredili podrobno analizo. Definirali smo načine uporabe, analizirali nekaj najbolj popularnih športov ter določili zahteve. S pomočjo dobre analize zahtev smo določili tri uporabniške vloge: administrator, lastnik športnega centra ter navaden uporabnik. Slika 4.1 prikazuje uporabniške vloge in akcije, ki jih sprožijo.

Navadni uporabnik ima možnost pregledovanja in iskanja dogodkov, s pri-



Slika 4.1: Prikaz vlog in akcij, ki jih lahko sprožijo. Administrator upravlja z vsemi akcijami, nato sledi lastnik športnega centra z že nekaj omejitvami. Najnižje je navaden uporabnik, ki lahko upravlja le z osnovnimi zahtevami.

javo v aplikacijo lahko dodaja lastne dogodke in se prijavlja na že obstoječe. Dodali smo še možnost, da lahko sami prijavimo druge uporabnike na lasten dogodek, pregled prijav ter pošiljanje obvestil.

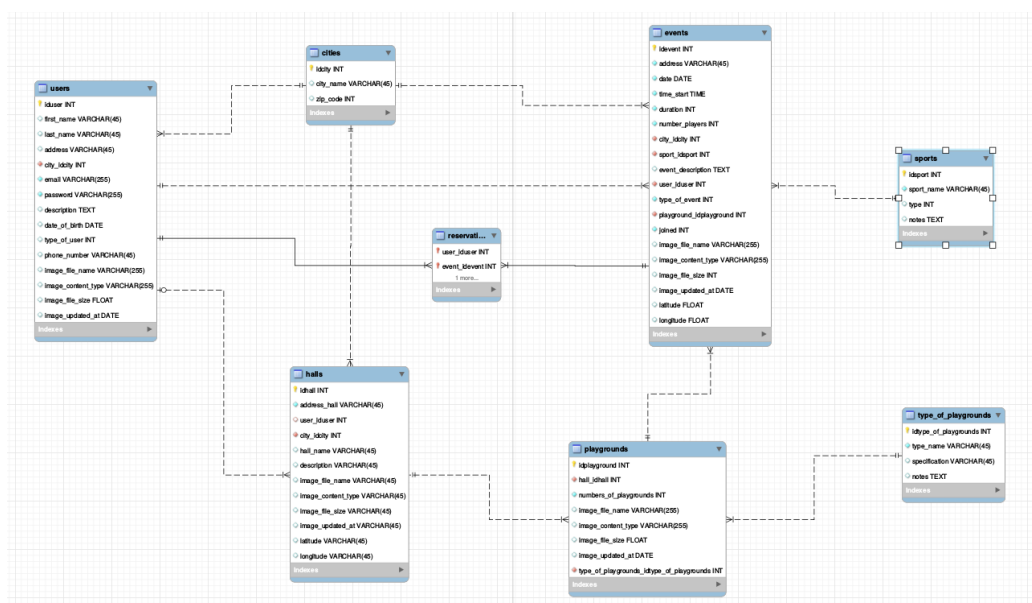
Vloga lastnika športnega centra je vloga, za odobritev katere je potrebno zaprositi administratorja. Ta vloga vsebuje vse funkcionalnosti kot prejšnja vloga, vendar z manjšimi spremembami. Omogoča dodajanje športnih dvoran (na podlagi teh dvoran je možno ustvariti dogodke).

Administrator je v naši aplikaciji vloga za nadzor in ima neomejen dostop. Dostopa do vseh dogodkov in uporabnikov ter jih lahko ureja in po potrebi izbriše. Ustvarja lahko dogodke kot navaden uporabnik ali kot lastnik športnega centra.

Vsi uporabniki spleta lahko pregledujejo dogodke. Za vse ostale funkcionalnosti pa se je potrebno registrirati.

4.2 Načrtovanje podatkovne baze

Na podlagi zgoraj opisanih funkcionalnosti smo zgradili načrt podatkovne baze (slika 4.2). Za gradnjo smo uporabili orodje MySQL Workbench. S pomočjo grafičnega vmesnika smo bazo hitro in enostavno zgradili. Model baze, ki smo ga zgradili, nam je služil samo kot načrt za lažjo implementacijo naše spletne aplikacije.



Slika 4.2: Struktura podatkovne baze, načrtane v MySQLWorkbench.

Bazo sestavlja 10 entitet, ki se med seboj povezujejo s primarnimi in tujimi ključi. Entitete po abecednem vrstnem redu so:

- cities - podatki o krajih in poštah številkah. V bazo jih lahko doda samo administrator.
- events - podatki o dogodku. Vsebuje tudi tuji ključ playground.id, ki je opcijski. Uporablja se za kreacijo dogodka, ki je vezan na športni center. Preko tega atributa aplikacija pridobi podatke o naslovu športnega centra.
- halls - podatki o športnih dvorinah.

- `playgrounds` - vsaka dvorana oz. športni center ima več različnih igrišč (za tenis, nogomet, košarko, odbojko, ...). Entiteta hrani podatke o vseh igriščih..
- `reservations` - hrani podatke o rezervacijah. Je vmesna tabela, ki povezuje tabelo `users` s tabelo `events`. Ob vsaki prijavi ali odjavi od dogodka se zgodi akcija, ki zapiše podatek v tabelo.
- `schema_migrations` - tabela, ki jo potrebuje Rails za migracijo skript v podatkovno bazo.
- `sports` - podatki o različnih športih. V bazo jih lahko doda samo administrator.
- `type_of_playgrounds` - podatki o tipu igrišča. Na primer: igrišče za tenis ima travnato površino, dvorana ima parket, in podobno. Tabela je bila uvedena zaradi normalizacije podatkovne baze.
- `users` - podatki o uporabnikih. Vloge se dodelijo po atributu `type_of_user`.

Ruby je objektno usmerjen programski jezik. Za dostopanje do podatkov uporablja tehniko, imenovano ORM. Je programska tehnika, ki se uporablja za pretvarjanje podatkov iz podatkovne baze v objekte v nekem objektnem jeziku [18]. Entitete so predstavljene kot modeli in vsebujejo vse attribute. Pri poizvedovanju se prenesejo vsi atributi, programer pa določi attribute za izpis. Ta tehnika je prijazna do programerja, pisanje poizvedb je enostavno in hitro s kratkimi ukazi. Vendar ni vse tako lepo. ORM postane pri ogromni količini podatkov počasen. Čeprav potrebujemo samo en atribut iz entitete, se prenese celotna zgradba. Primerna je za manjše in mikro projekte z malo entitetami, atributi ter vnešenimi podatki.

Poglavje 5

Implementacija

V tem poglavju se bomo osredotočili na samo aplikacijo. Predstavljen bo celoten proces razvoja. Postregli bomo z izseki programske kode, zaslonskimi slikami ter opisi.

5.1 Postavitev ogrodja in konfiguracija

Da smo lahko začeli razvijati našo spletno aplikacijo, je bilo potrebno zgenerirati programsko ogrodje. To smo storili z ukazom:

```
rails new Sportnik
```

Povezavo do podatkovne baze smo vzpostavili v datoteki `database.yml`, ki se nahaja v mapi `Sportnik/config/`. Aplikacijo smo povezali z zunanjim strežnikom, na katerem teče podatkovna baza MySQL. Koda, ki ustvari povezavo (povezava je ustvarjena samo za razvoj (*development*)):

```
development:
  adapter: mysql2
  encoding: utf8
  pool: 5
  username: lovro
  password: lovro123
```

```
host: nissil.si  
database: test  
port: 3306
```

V Rails imamo več različnih ukazov, s katerimi komuniciramo s podatkovno bazo:

- `rake db:create` - ustvari podatkovno bazo,
- `rake db:migrate` - migracija podatkovne skripte v podatkovno bazo,
- `rake db:rollback` - izbriši zadnjo migracijo v podatkovni bazi,
- `rake db:drop` - izbriši podatkovno bazo.

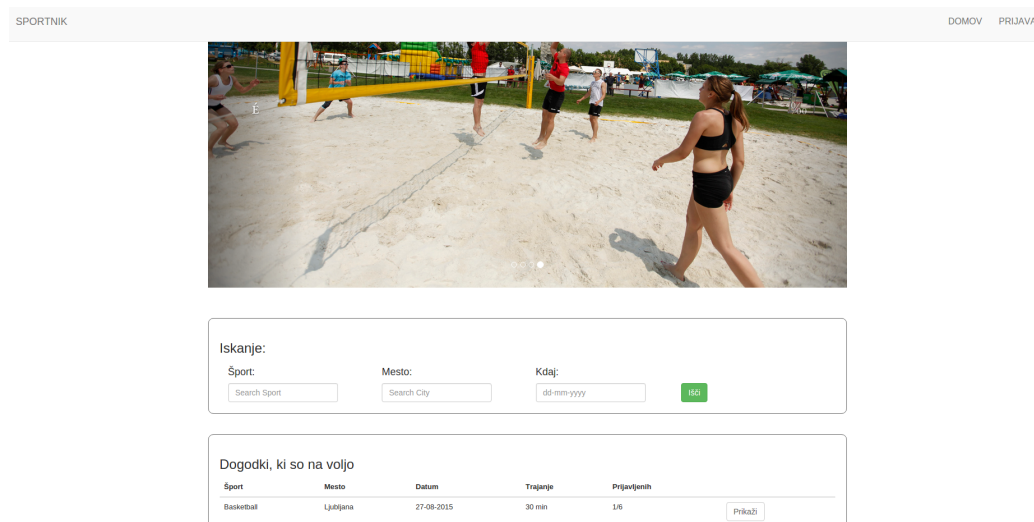
Ko smo pripravili vse potrebno, lahko začnemo z razvojem. Za kreacijo migracijskih skript, modelov, pogledov in kontrolerjev smo uporabili ukaz **rails g scaffold**. Je zelo enostaven za uporabo, vendar vsebuje veliko stvari, ki jih ne potrebujemo. Primer generiranja modela City:

```
rails g scaffold City name:string zip_code:integer
```

5.2 Opis uporabniškega vmesnika

Ob prvem dostopu do aplikacij se odpre osnovni pogled. Do njega lahko dostopajo tako prijavljeni kot tudi neprijavljeni uporabniki. Prikaže se dia-projekcija slik, ki se samodejno spreminja. Pod njimi je okno za iskanje po določenih parametrih (po športu, kraju in datumu). Pod iskalnikom pa se nahajajo vsi aktualni dogodki oziroma vsi najdeni aktualni dogodki, če je uporabnik uporabil iskalnik. Osnoven pogled prikazuje slika 5.1.

Poleg vsakega dogodka se nahaja gumb "Prikaži". Namenjen je prikazu podrobnosti določenega dogodka. Podrobnosti o dogodku so vidne samo registriranim uporabnikom, saj želimo, da se dogodka udeležijo le tisti, ki so se nanj prijavili. S tem se izognemo preveliki množici ljudi na dogodku,



The screenshot shows the main interface of the application. At the top, there is a navigation bar with the text "SPORTNIK" on the left and "DOMOV" and "PRIJAVA" on the right. Below the navigation bar is a large image of a beach volleyball court with several players. Underneath the image is a search form titled "Iskanje:". The form has three input fields: "Šport:" with a placeholder "Search Sport", "Mesto:" with a placeholder "Search City", and "Kdaj:" with a placeholder "dd-mm-yyyy". There is a green "Išči" button to the right of the date field. Below the search form is a table titled "Dogodki, ki so na voljo". The table has five columns: "Šport", "Mesto", "Datum", "Trajanje", and "Prijavljenih". The table contains one row of data: "Basketball", "Ljubljana", "27-08-2015", "30 min", and "1/6". There is a "Prikaži" button to the right of the table.

| Šport | Mesto | Datum | Trajanje | Prijavljenih |
|------------|-----------|------------|----------|--------------|
| Basketball | Ljubljana | 27-08-2015 | 30 min | 1/6 |

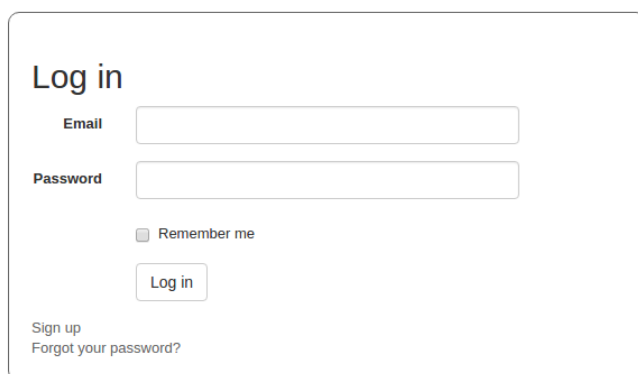
Slika 5.1: Osnoven pogled naše aplikacije.

organizatorji pa imajo nadzor nad tem, ali je dovolj udeležencev, da bi lahko dogodek izpeljali.

Če dostopamo kot neregistriran uporabnik in bi se želeli udeležiti športnega dogodka, nam aplikacija ob kliku na gumb "Prikaži" ponudi prijavo oziroma registracijo, če uporabnik še ni registriran. Slika 5.2 prikazuje okno, kjer se lahko uporabnik prijavi oziroma registrira. Prav tako nam pritisk na gumb "Prijava" v navigacijski vrstici ponudi enako okno.

5.2.1 Navaden uporabnik

Ob prijavi je osnovno okno enako kot pri neprijavljenem uporabniku. Razlika je v navigacijski vrstici, kjer so gumbi z dodatnimi funkcionalnostmi. Dodatne funkcionalnosti, ki jih ima na voljo prijavljen uporabnik, so: pregled dvoran oz. športnih centrov, dodajanje lastnega dogodka ter spustni meni z nastavitvami (urejanje profila, pregled lastnih dogodkov in prijav na dogodke ter zgodovina (slika 5.3)).



Log in

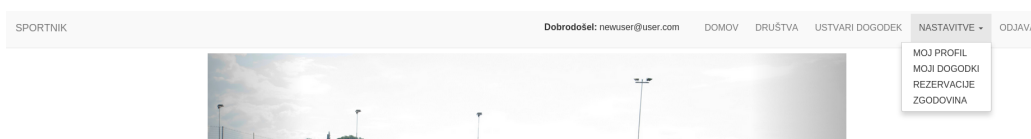
Email

Password

☐ Remember me

[Sign up](#)
[Forgot your password?](#)

Slika 5.2: Okno za prijavo oziroma registracijo uporabnika.



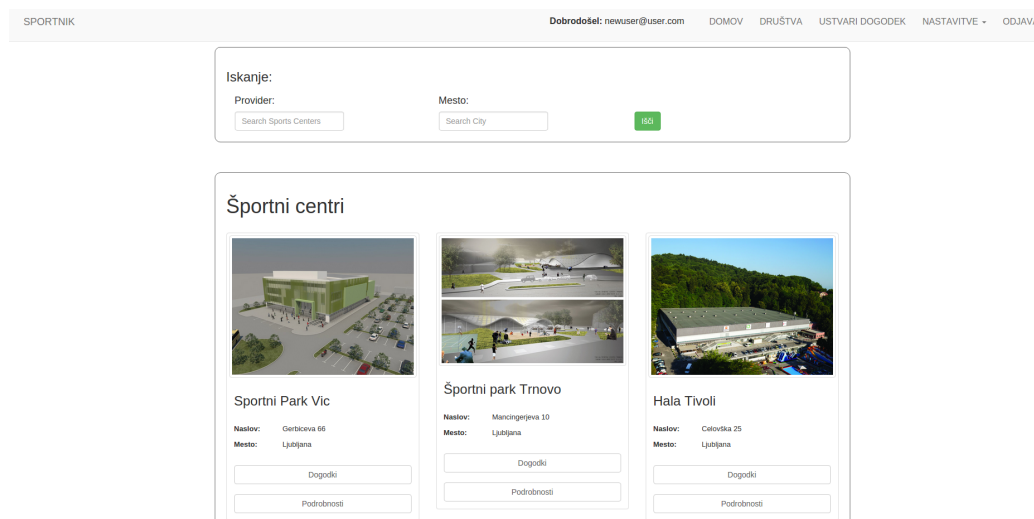
Slika 5.3: Navigacijska vrstica navadnega uporabnika.

Seznam športnih centrov

Če uporabnik klikne na funkcijo "Društva", se pojavi množica tablic (slika 5.4), na katerih so vsi vpisani športni centri. Na vsaki tablici je slika športne dvorane, pod njo pa se nahajata dva gumba - "Dogodki" in "Podrobnosti". Prvi nam prikaže vse aktualne dogodke, ki so na voljo, drugi pa informacije o tem športnem centru. Nad seznamom športnih centrov se nahaja iskalno okno, ki omogoča iskanje športnega centra po imenu ali mestu. Če iskalnik ne najde zadetka, izpiše obvestilo.

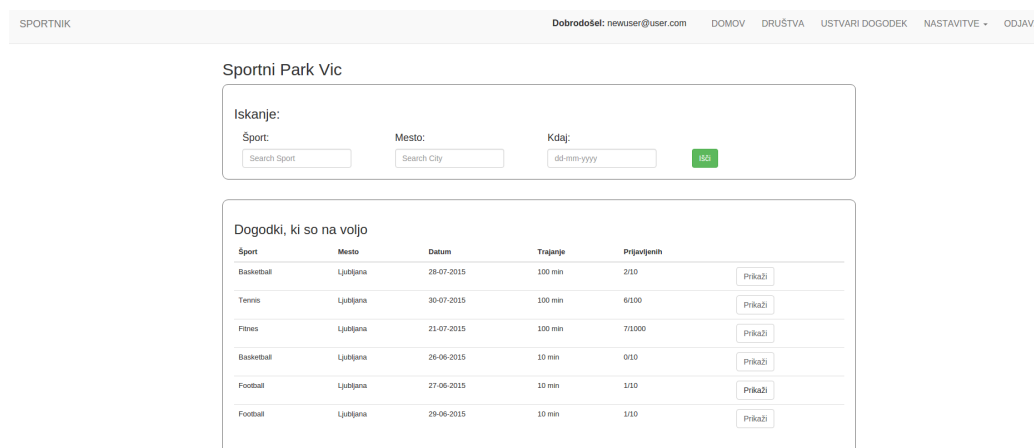
Prikaz športnega centra

Ob kliku na gumb "Dogodki" se pojavi okno, v njem pa se nahajajo vsi aktualni dogodki. Okno je enako tistemu na domači strani, razlikuje se samo po vsebini. Drugi gumb "Podrobnosti", pa nam prikaže okno (slika 5.5), ki



Slika 5.4: Prikaz vseh športnih centrov.

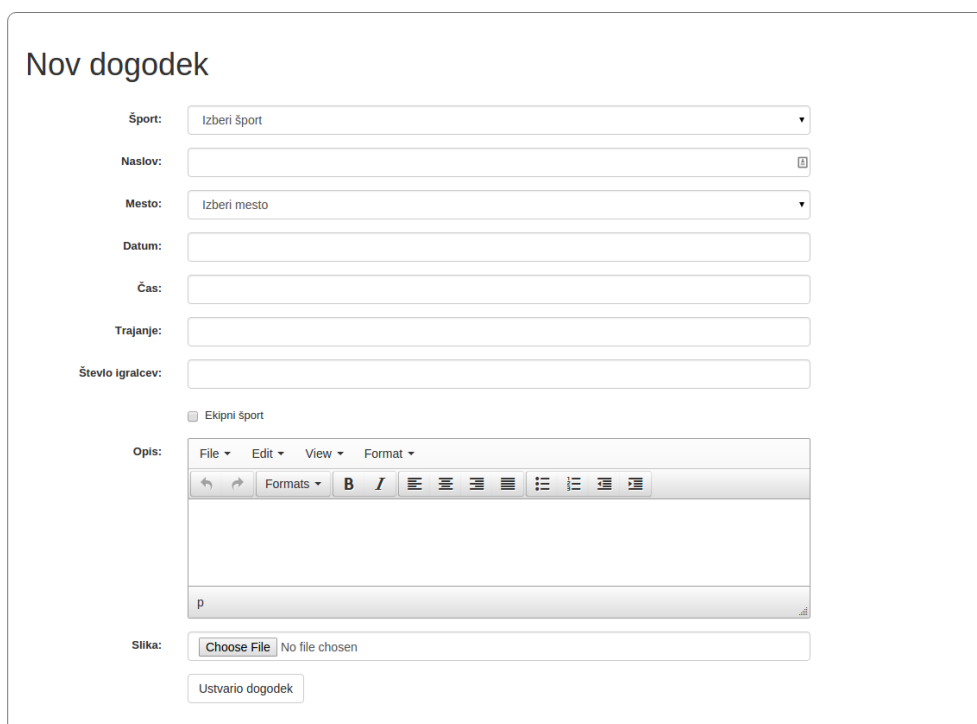
predstavlja podrobnejši opis športnega centra. Tukaj so podatki o naslovu (zraven je tudi zemljevid), številu igrišč športnega centra, predstavitvena slika ter opis. Če ima športni center dodana športna igrišča, se pod zemljevidom nahaja gumb "Prikaži igrišča". Ob kliku na gumb se pojavi seznam z vsemi vnešenimi igrišči. Za prikaz skrbi AJAX.



Slika 5.5: Dogodki, ki pripadajo športnemu centru.

Dodajanje dogodka

Funkcija za vnos dogodka se nahaja v navigacijski vrstici pod gumbom "Dodaj dogodek". Prikaže se nam obrazec, ki zahteva vnos naslednjih podatkov (slika 5.6):



Slika 5.6: Obrazec za kreacijo dogodka.

- Šport - meni z izbiro športa. Šport lahko doda samo administrator (zaradi različnih specifikacij pri posameznem športu).
- Naslov in mesto - kraj, kjer se bo dogodek odvijal. Na podlagi naslova se izračunajo parametri za izris Google zemljevida.
- Datum in čas - jQuery gradnika za izbiro.
- Trajanje - vnosno polje, kjer lahko vpišemo samo cela števila. Čas trajanja se vnaša v minutah.

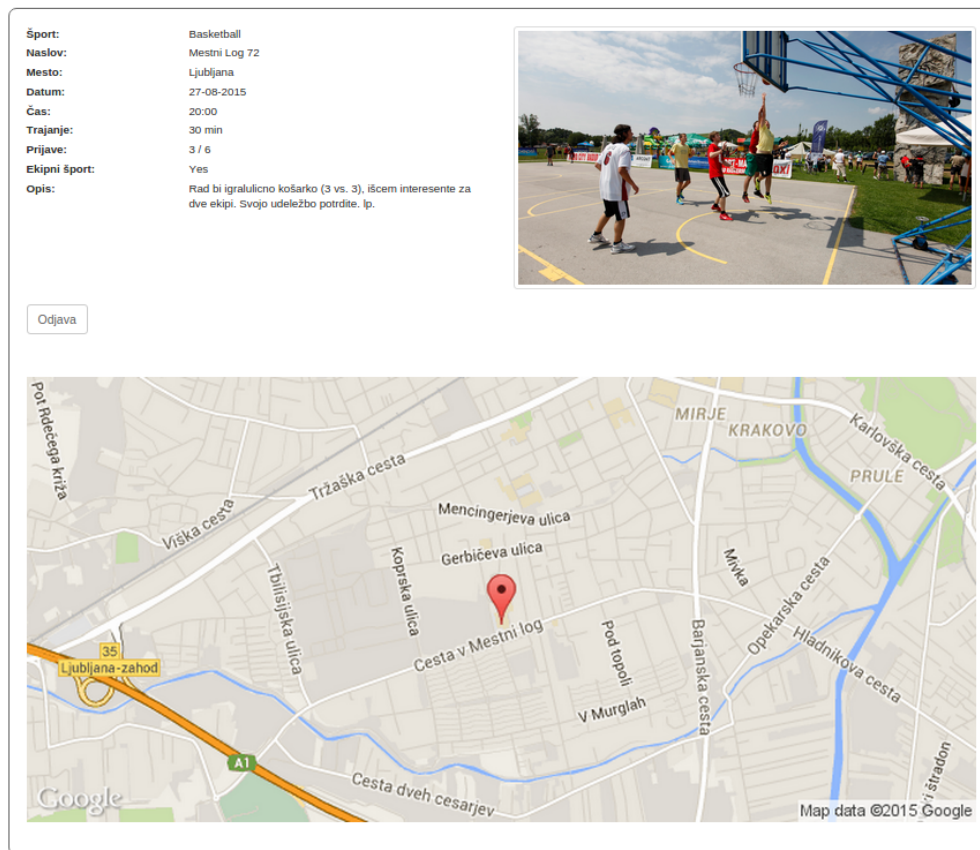
- Število udeležencev - koliko udeležencev se lahko prijavi na dogodek. Dovoljen je vnos samo celih števil.
- Ekipni šport - izbirni meni nam omogoča izbor, ali je športna dejavnost ekipni ali individualni šport.
- Opis - vnosno polje za opis športne dejavnosti. Uporabili smo urejevalnik besedila TinyMCE.
- Slika - vnos slike za lažjo vizualizacijo.
- Ustvari dogodek - v modelu Event se preveri pravilnost vseh vnešenih podatkov. Če vse ustreza zahtevam, se dogodek ustrezno zabeleži v podatkovno bazo. V nasprotnem primeru se ob poljih, ki so bila napačno izpolnjena, izpiše obvestilo (slika 5.7).



Slika 5.7: Primer obvestila, če smo pozabili izpolniti vnosno polje.

Prikaz dogodka

Če želimo videti podrobnosti o dogodku, enostavno kliknemo na gumb "Prikaži", ki se nahaja poleg dogodka. Prikažejo se podatki o športu in predstavitevna slika (slika 5.8). S pomočjo Google zemljevida se poišče lokacijo, kjer se bo ta dogodek izvajal. Gumb "Prijava" omogoči avtomatsko prijavo na dogodek. Po prijavi dobimo obvestilo o uspešni prijavi. Spremeni se podatek o številu udeležencev, pojavi se Gumb "Odjava". Če smo se na dogodek prijavili pomotoma, se ob kliku na odjavo podatki posodobijo na prejšnje stanje in dobimo obvestilo o odjavi.



Slika 5.8: Prikaz podrobnosti o športnem dogodku.

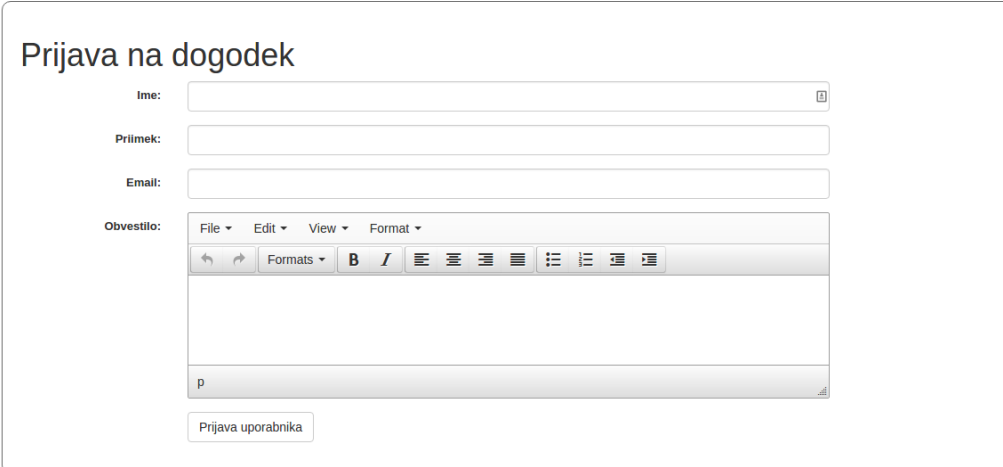
Dodajanje uporabnika na dogodek in brisanje z dogodka

Če smo dogodek ustvarili sami, imamo na voljo še dve dodatni funkcionalnosti. To sta pregled vseh prijav na dogodek ter možnost prijave drugih uporabnikov. Gumba se prikažeta poleg gumba "Odjava" ali "Prijava" (odvisno, ali smo prijavljeni na dogodek ali ne).

Prvi gumb "Preglej rezervacije" nam prikaže okno, v katerem je seznam vseh uporabnikov, ki so se prijavili na ta dogodek. Podatki so: ime, priimek in elektronski naslov. Ob imenu vsakega uporabnika se nahaja tudi gumb "Izbriši". Ta gumb omogoča organizatorju dogodka odjaviti uporabnika, če le-ta to želi. Uporabniku se avtomatično pošlje obvestilo o odjavi z dogodka.

Drugi gumb "Prijava drugega uporabnika" nam prikaže obrazec (slika

5.9). Vnosna polja so: ime, priimek, elektronski naslov in obvestilo. Polja ime, priimek in obvestilo so opcijska, ne vplivajo na validacijo pri prijavi, ampak so namenjena generiranju elektronske pošte. Na prijavo vpliva samo polje elektronski naslov. V bazi se preveri, ali vpisani uporabnik obstaja, naredi se rezervacija in pošlje obvestilo. Ne moremo pa prijaviti uporabnika, ki še ni registriran v aplikacijo.



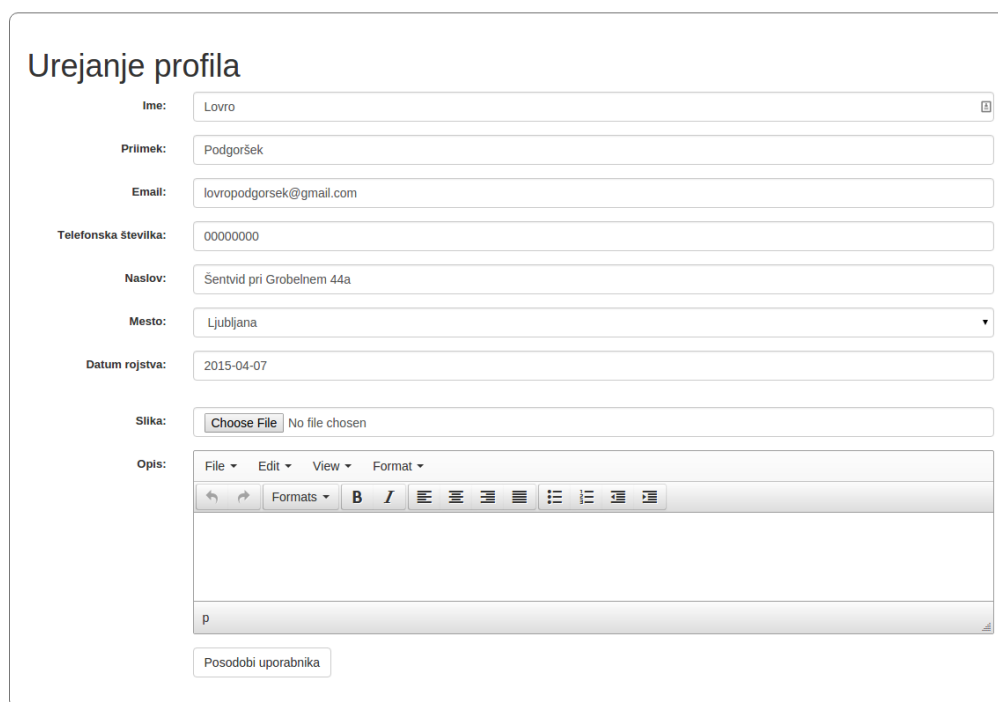
Slika 5.9: Obrazec za prijavo drugega uporabnika.

Nastavitve in ostale možnosti

V meniju "Nastavitve" se nahajajo funkcije: pregled lastnega profila, pregled lastnih rezervacij na aktualnih dogodkih, pregled lastnih dogodkov ter zgodovina (prikaz že obiskanih dogodkov). Ob kliku na profil se nam prikaže stran z našimi podatki ter sliko. Če smo se samo registrirali, imamo v podatkih samo elektronski naslov in mesto. Zaradi težav pri implementaciji smo določili, da se mesto samodejno nastavi na Ljubljano. Na voljo je gumb "Uredi profil", ki nam ponudi obrazec za urejanje (slika 5.10). Ob kliku na gumb "Posodobi uporabnika" se vnešeni podatki shranijo v podatkovno bazo.

Za predstavitev rezervacij, lastnih dogodkov in zgodovine uporabljamo tabele. Dogodki so predstavljeni kot vrstice v tabeli. Ob rezervaciji je ob

vsakem dogodku na voljo še gumb "Odjava". Pri pregledu lastnih dogodkov smo dodali gumb "Pregled prijav" ter gumbe za urejanje.



Urejanje profila

Ime: Lovro

Priimek: Podgoršek

Email: lovropodgorsek@gmail.com

Telefonska številka: 00000000

Naslov: Šentvid pri Grobelnem 44a

Mesto: Ljubljana

Datum rojstva: 2015-04-07

Slika: Choose File No file chosen

Opis: File Edit View Format
Formats B I [text alignment icons] [list icons]
p

Posodobi uporabnika

Slika 5.10: Obrazec za urejanje profila.

5.2.2 Lastnik športnega centra oziroma Ponudnik

Uporabniško vlogo "Lastnik športnega centra" oziroma "Ponudnik" bodo imeli vsi tisti uporabniki, ki imajo v lasti športne dvorane in igrišča. Potrebno bo poslati prošnjo in dokazila administratorju, ta pa jim bo odobril pravice. Kot ponudniki imamo možnost uporabe enakih funkcionalnosti kot navaden uporabnik, le z manjšimi razlikami: drugačen obrazec za vnos športnega dogodka ter možnost dodajanja športnih dvoran in igrišč.

Ko bo nov ponudnik prvič dostopal do naše aplikacije, bo moral v sistem najprej vnesti športno dvorano in igrišča. Šele nato se mu bo v navigacijski vrstici ponudila funkcija za dodajanje športnih dogodkov. Namesto

gumba "Ustvari dogodek" bo imel možnost "Dodaj športni center" ali "Dodaj igrišče".

Dodajanje dogodka

Ob kliku na gumb "Ustvari dogodek" se nam odpre obrazec (slika 5.11), ki je podoben kot tisti, ki se odpre navadnemu uporabniku. Razlikuje se v vnosnem in potrditvenem polju:

The screenshot shows a web form titled "Nov dogodek". It contains the following elements:

- Form fields: Šport (dropdown), Dvorana (dropdown), Datum (text), Čas (text), Trajanje (text), Število igralcev (text).
- Checkboxes: ☐ Ekipni šport, ☐ Ustvari rezervacijo.
- Description field: A rich text editor with a menu (File, Edit, View, Format) and various formatting icons.
- Slika field: A text input with a "Choose File" button and the text "No file chosen".
- Submit button: "Ustvari dogodek".

Slika 5.11: Obrazec za ustvarjanje športnega dogodka v vlogi ponudnika.

- Dvorana - ponudnik izbere dvorano. Na podlagi tega atributa se prikaže novo vnosno polje za izbiro igrišč, ki so dodani tej dvorani. Glede na naslov dvorane se določi lokacija dogodka.
- Ustvari rezervacijo - potrditveno polje za odločanje, ali se kot ponudnik želimo udeležiti dogodka, ali ga želimo samo ustvariti. Če ustvarimo dogodek kot navaden uporabnik, se rezervacija naredi avtomatično.

Dodajanje športnih dvoran in igrišč

V meniju pod gumbom "Nastavitve" se nam dodata še dva gumba - "Dvorane" in "Igrišča". S pomočjo teh gumbov lahko dostopamo do seznama vseh dvoran in igrišč. Prikazane so v obliki seznama, ob vsaki pa se nahajajo še gumbi za prikaz, urejanje ter izbris (slika 5.12).

| Seznam športnih dvoran | | | | | |
|------------------------|------------------|-----------|---------|-------|---------|
| Športno društvo | Naslov | Mesto | | | |
| Sportni Park Vic | Gerbiceva 66 | Ljubljana | Prikaži | Uredi | Izbrisi |
| Sportni park Trnovo | Mancingerjeva 10 | Ljubljana | Prikaži | Uredi | Izbrisi |
| Hala Tivoli | Celovška 25 | Ljubljana | Prikaži | Uredi | Izbrisi |
| Sportni park Radomlje | Skrjancevo 19 | Ljubljana | Prikaži | Uredi | Izbrisi |
| Dodaj novo dvorano | | | | | |

Slika 5.12: Seznam športnih dvoran.

Ob kliku na gumb "Dodaj športno dvorano" se nam prikaže obrazec za vnos športne dvorane (slika 5.13). Obrazec vsebuje:

| | |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Nova dvorana | |
| Športno društvo: | <input type="text"/> |
| Naslov: | <input type="text"/> |
| Mesto: | <input type="text" value="Izberi mesto"/> |
| Opis: | <div><div>File Edit View Format</div><div>↶ ↷ Formats B <i>I</i> [align icons] [list icons] [table icon]</div><div><p>p</p></div></div> |
| Slika: | <input type="button" value="Choose File"/> No file chosen |
| <input type="button" value="Ustvari dvorano"/> | |

Slika 5.13: Obrazec za vnos športne dvorane.

- Športna dvorana - ime športnega centra ali društva.
- Naslov in mesto - iz naslova in mesta se izračunajo parametri za izris Google zemljevida.
- Opis - kratka predstavitev športnega centra.
- Slika - če ne dodamo slik, se uporabi nevtralna slika, ki je že vnešena v naši aplikaciji.

Ob uspešnem vnosu nove športne dvorane je potrebno dodati še igrišča. V nasprotnem primeru ne moremo ustvariti dogodka, ki bi se izvajal v tej dvorani. To naredimo s klikom na "Nastavitve" in nato "Igrišča". Prikaže se nam seznam z vsemi igrišči, ki smo jih že vnesli (seznam je enak tistemu na sliki 5.12). Na dnu seznama imamo gumb "Dodaj novo igrišče", ki nam prikaže obrazec (slika 5.14). Izberemo dvorano, kateri pripada. Da ne bi imeli v bazi podvojenih tipov igrišč, smo dodajanje omogočili le administratorju. Ponudnik pa lahko izbere, kakšne vrste igrišče je (travnato, samo za tenis, bazen, ...). V polju "Oznaka igrišča" vnesemo številko oziroma oznako igrišča, kjer je na koncu še možno dodati sliko. S pomočjo slike si lahko uporabniki predstavljajo, kakšno je ponujeno igrišče.

Dodaj igrišče

Dvorana: Izberi dvorano

Tip igrišča: Izberi tip igrišča

Oznaka igrišča:

Slika: Choose File No file chosen

Ustvari igrišče

Slika 5.14: Obrazec za vnos igrišča.

5.2.3 Administrator

Kot smo že omenili, ima administrator nadzor nad celotno aplikacijo. To je vloga brez omejitev. Poleg že vseh predstavljenih funkcionalnosti ima še nekaj dodatnih, kot so: pregled nad uporabniki in upravljanje z njimi, vnos športov, mest in tipov igrišč. Možnost kreiranja dogodkov kot navaden uporabnik ali ponudnik, možnost kreiranja dogodka pri vseh športnih centrih in drugo.

Tako kot navaden uporabnik in lastnik športnih centrov ima tudi administrator svojo navigacijsko vrstico. Na njej so gumbi, ki mu omogočajo vso funkcionalnost. Pojavi se tudi nov gumb "Administrator" v obliki spustnega menija. V njem lahko dostopa do pregleda uporabnikov, športov, mest in tipov igrišč.

Pregled uporabnikov

Pogledali si bomo funkcionalnost, ki se nahaja pod možnostjo "Pregled uporabnikov". Vidimo lahko seznam vseh registriranih uporabnikov ter iskalno okno (slika 5.15). Za vsakega uporabnika vidimo podatke o imenu, priimku, elektronski pošti, mestu in pravicah. Ob vsakem uporabniku imamo še gumbe:

- "Prikaži" - dostop do uporabniškega profila. Profil lahko samo pregledujemo, če nas zanimajo podrobnosti o uporabniku. Ob dostopu do lastnega profila imamo še gumb "Uredi".
- "Dodaj pravice" - vsakemu uporabniku poljubno nastavimo pravice.
- "Izbriši" - izbriše uporabnika iz baze. Izbrišejo se vsi njegovi dogodki, rezervacije in zgodovina. Pred dokončnim izbrisom se pojavi še varnostno obvestilo, ki zahteva potrditev dokončnega izbrisa uporabnika (slika 5.16).

Za funkcionalnost dodajanje športov, mest in tipov igrišč je postopek enak kot pri pregledu uporabnikov. Ustvarili smo obrazce za vnos, ki pa so

Iskanje:

Ime:

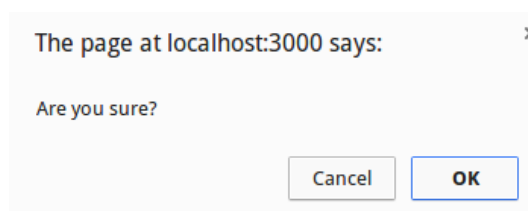
Priimek:

Email:

Seznam uporabnikov

| Ime | Priimek | Email | Pravice | Mesto | | | |
|----------|-----------|--------------------------|----------------|-----------|----------------------------------------|----------------------------------------------|----------------------------------------|
| Lovro | Podgoršek | lovropodgorsek@gmail.com | Administration | Ljubljana | <input type="button" value="Prikaži"/> | <input type="button" value="Dodaj pravice"/> | <input type="button" value="Izbriši"/> |
| Provider | Provider | provider@provider.com | Provider | Ljubljana | <input type="button" value="Prikaži"/> | <input type="button" value="Dodaj pravice"/> | <input type="button" value="Izbriši"/> |
| Test | Test | test@test.com | Only User | Ljubljana | <input type="button" value="Prikaži"/> | <input type="button" value="Dodaj pravice"/> | <input type="button" value="Izbriši"/> |

Slika 5.15: Prikaz seznama vseh uporabnikov.



Slika 5.16: Obvestilo ob izbrisu uporabnika.

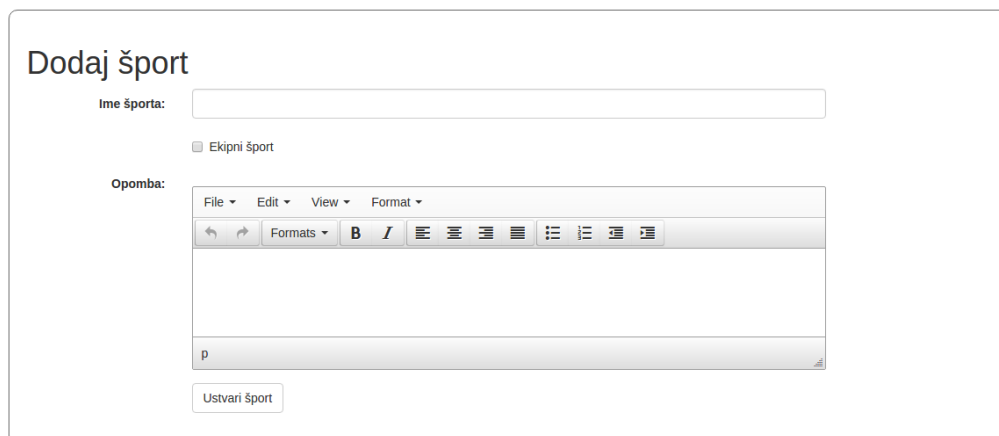
prilagojeni potrebam za vnašanje (slika 5.17 oziroma slika 5.18).

5.3 Funkcionalnosti

V tem razdelku bomo izpostavili še nekaj funkcionalnosti, ki smo jih razvili in vplivajo na dobro uporabniško izkušnjo. Predstavili jih bomo z opisom, slikami in programsko kodo.

5.3.1 Pošiljanje elektronske pošte

Kot smo omenili že pri prejšnjih poglavjih, uporabnika za vsako dejanje obvestimo. Za obveščanje smo si zaradi enostavne in brezplačne uporabe izbrali



Slika 5.17: Obrazec za vnos športa.



Slika 5.18: Obrazec za vnos mesta.

elektronsko pošto.

Za pošiljanje elektronske pošte potrebujemo zunanji strežnik, ki pošilja elektronsko pošto. Izbrali smo API Mandrill. To je spletna storitev, ki ga je razvilo podjetje MailChimp leta 2012. Leta 2010 je obstajala samo ideja, sedaj pa je to največja storitev za elektronsko pošto in ima več kot 500 000 aktivnih strank. Uporablja se za pošiljanje ponastavljenih gesel, pozdravnih obvestil, marketinških elektronskih pošt in drugih podobnih sporočil.

Potrebno se je bilo registrirati na spletni strani (<https://mandrillapp.com>), nato pa smo morali med nastavitvami poiskati podatke: gostiteljev naslov (*host*), vrata (*port*), uporabniško ime ter geslo. Te podatke potrebujemo za dostopanje do Mandrill-ovega strežnika. V aplikaciji smo zgenerirali model `User_mailer` z ukazom:

```
rails g mailer user_mailer
```

V datotekah */config/environments/development.br* in */config/environments/production.br* smo dodali kodo za povezavo do strežnika:

```
config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  address: 'smtp.mandrillapp.com',
  port: 587,
  authentication: 'plain',
  enable_starttls_auto: true,
  user_name: 'lovropodgorsek@gmail.com',
  password: 'FP5cz_6etKMpGCD6oTG5XA'
}
```

Potrebno je bilo še zgenerirati vsebino elektronske pošte. V datoteki */app/mailers/user_mailer.rb* so metode, ki pripravijo podatke, potrebne za generiranje vsebine elektronske pošte. Primer metode za rezervacijo dogodka:

```
def notice_of_reservation(user, event, notes, provider_id)
  @user = user
  @event = event
  @notes = notes
  @provider = User.find(provider_id)
  @sport = Sport.find(@event.sport_id)
  @city = City.find(@event.city_id)

  mail to: @user.email, subject: 'Notice of Reservation'
end
```

Zgeneriramo še elektronsko pošto iz podatkov, ki smo si jih pripravili s pomočjo zgoraj opisane metode. Sestavljen je iz HTML značk in dinamičnih podatkov. HTML datoteka se nahaja na lokaciji */app/view/user.html.erb*.

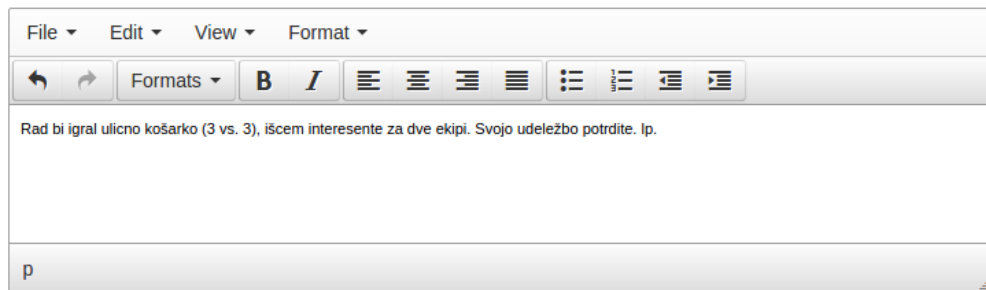
Tako smo zgenerirali elektronsko pošto za nekega poljubnega uporabnika. Da pa lahko to elektronsko pošto pošljemo uporabniku, je potrebno dodati še akcijo v kontrolerju, ki sproži celoten postopek. Spodaj je prikazana koda za pošiljanje:

```
UserMailer.notice_of_reservation(@user, event,  
event_params[:description], current_user.id).deliver_now
```

5.3.2 TinyMCE

Uporaba HTML elementa `<textarea>` ni najprimernejša za vnašanje daljšega teksta v spletnih aplikacijah. V ta namen smo se odločili, da za vnosno polje "Opis" uporabimo TinyMCE urejevalnik besedila. Je odprtokodna programska oprema, ki deluje neodvisno od platforme. Deluje na osnovi HTML in JavaScript-a.

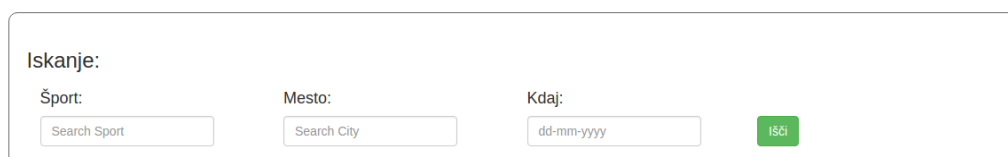
Za implementacijo TinyMCE urejevalnika v našo aplikacijo smo dodali knjižnico v datoteko `GemFile.rb` ter sledili navodilom za implementacijo. Koda je kratka in razdrobljena po večih datotekah, zato je ne bomo podrobneje obravnavali. TinyMCE urejevalnik besedila je predstavljen na sliki 5.19.



Slika 5.19: TinyMCE urejevalnik besedila.

5.3.3 Iskanje dogodkov

Uporabniki običajno iščejo le tisto, kar jih zanima. V ta namen smo implementirali iskalnik dogodkov (slika 5.20). Vstavili smo ga pri vseh seznamih, ki so dovolj veliki, da ga potrebujejo, ter jim ustrezno prilagodili iskalne parametre.



Slika 5.20: Iskalnik dogodkov.

Za iskanje smo uporabili tri najbolj pomembne parametre, ki bi uporabnika zanimali. To so šport, kraj dogodka in datum. Razporedili smo jih po pomembnosti. Najpomembnejši parameter je "Šport", saj ne želimo pregledovati seznama s športi, ki nas ne zanimajo. Sledi "Mesto", v katerem kraju se bo odvijal športni dogodek ter na koncu še "Datum" za iskanje po datumu. Iskanje postane še bolj specifično, če vnesemo več parametrov hkrati.

Iskalno okno je zgrajeno iz HTML značk, ki pa vsebujejo Rails kodo za dinamičnost. Ob kliku na gumb "Išči" se parametri pošljejo na strežnik. V kontrolerju se pregledajo parametri ter pošljejo poizvedbo v podatkovno bazo. Spodaj je prikazana koda v kontrolerju, ki je odgovorna za proženje poizvedb:

```
if params[:search_date] != nil
  if params[:search_sport] != '' and
    params[:search_city] != '' and
    params[:search_date] != ''
    @sports = Sport.search_name(params[:search_sport])
    @cities = City.search_name(params[:search_city])
    @events = Event.where(sport_id: @sports.map(&:id),
      city_id: @cities.map(&:id), :date => params[:search_date].to_date)
```

```
elsif params[:search_sport] != '' and params[:search_date] != ''
  @sports = Sport.search_name(params[:search_sport])
  @cities = City.search_name(params[:search_city])
  @events = Event.where(sport_id: @sports.map(&:id),
    city_id: @cities.map(&:id))
elsif params[:search_sport] != '' and params[:search_date] != ''
  @sports = Sport.search_name(params[:search_sport])
  @events = Event.where(sport_id: @sports.map(&:id),
    :date => params[:search_date].to_date)
elsif params[:search_city] != '' and
params[:search_date] != ''
  @cities = City.search_name(params[:search_city])
  @events = Event.where(city_id: @cities.map(&:id),
    :date => params[:search_date].to_date)
elsif params[:search_sport] != ''
  @sports = Sport.search_name(params[:search_sport])
  @events = Event.where(sport_id: @sports.map(&:id))
elsif params[:search_city] != ''
  @cities = City.search_name(params[:search_city])
  @events = Event.where(city_id: @cities.map(&:id))
elsif params[:search_date] != ''
  @events = Event.where(:date => params[:search_date].to_date)
else
  @events = Event.all
end
else
  @events = Event.all
end
```


5.4 Postavitev strežnika in zagon aplikacije

Našo spletno aplikacijo smo postavili na zasebni strežnik, na katerem tečejo Nginx, Ruby on Rails in Unicorn [6]. Bazo pa smo postavili na drugem strežniku (baza MySQL). Strežnik smo zakupili na spletni strani

<https://www.digitalocean.com>, ima pa že določene specifikacije za lažjo postavitev. Za postavitev smo uporabili kombinacijo Nginx [17] in Unicorn-a. Nginx je spletni strežnik, ki je nekakšna alternativa popularnemu spletnemu strežniku Apache. Je hiter, zmožen dela z veliko hkratnimi povezavami ter ne porabi veliko sistemskih sredstev. Unicorn pa je spletni strežnik za Rack (Ruby-ev vmesnik za spletne strežnike) aplikacije, ki omogoča hitre obdelave HTTP glave in procesiranje Ruby zahtevkov.

Poglavje 6

Sklepne ugotovitve

V diplomskem delu smo prikazali celoten razvoj spletne aplikacije za organizacijo športnih dogodkov. Najprej smo predstavili orodja in jezike, ki smo jih uporabili pri razvoju. Z analizo smo zajeli začetne funkcionalnosti, ki pa smo jih med samo implementacijo še dopolnjevali in dodajali. Na podlagi analize smo zgradili podatkovni model, ki nam je bil v oporo pri sami implementaciji. Na koncu pa smo aplikacijo postavili na lastni strežnik.

Aplikacija ima preprost in minimalističen uporabniški vmesnik. Primerna je za vse starostne skupine, do želenih funkcionalnosti pa lahko dostopamo preko navigacijske vrstice. Vmesnik vsebuje samo informacije, ki so najnujnejše, ker smo se držali načela minimalističnosti (manj je več). S pomočjo Bootstrap-a pa se vmesnik prilagaja na različne širine zaslonov naprav, kar je še posebej razveseljujoče za vse uporabnike mobilnih naprav, saj jim to omogoča uporabo aplikacije tudi brez prenosnega ali pa namiznega računalnika.

Trenutno je aplikacija še v stanju testiranja, zato še ni primerna za množično uporabo. Potrebno je še testiranje na realnih problemih (kako se odziva pri večjem številu uporabnikov, ob velikem številu športnih dogodkov, ob uporabi večjega števila športnih centrov in igrišč). S pomočjo odziva uporabnikov se bodo lažje odpravile napake, ki smo jih naredili pri sami implementaciji.

6.1 Nadaljnje delo

Poleg popravljanja napak pa imamo na voljo še dovolj prostora za nadaljnji razvoj:

- Grupiranje uporabnikov v športne ekipe - uporabniki so lahko člani ekipe. Z eno samo prijavo bi lahko prijavili celotno ekipo.
- Komunikacija z uporabniki znotraj aplikacije - pošiljanje osebnih sporočil znotraj ekipe.
- Kreiranje športnega turnirja - ponudnikom dodati možnost organizacije celotnega turnirja.
- Elektronsko plačevanje prijavnine - športno dvorano je potrebno plačati. Plačilo se bo lahko naredilo kar od doma.
- Ocenjevanje uporabnikov - glede na šport bi se lahko ocenjevalo in rangiralo posamezne uporabnike, na primer vse uporabnike, ki igrajo tenis.

Ko bo aplikacija prišla iz razvojnega v izvajalno okolje oziroma v množično uporabo, bomo postopoma vpeljevali nadgradnje, obenem pa skrbeli za vzdrževanje. Izdelava spletne aplikacije v okviru diplomskega dela je realizirana v skladu z zahtevami, pričakovanji in cilji.

Literatura

- [1] Ajax. [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)). Dostopal: 09-08-2015.
- [2] Bootstrap (front-end framework). [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)). Dostopal: 11-08-2015.
- [3] CSS. https://en.wikipedia.org/wiki/Cascading_Style_Sheets. Dostopal: 09-08-2015.
- [4] Git (software). [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software)). Dostopal: 11-08-2015.
- [5] GitHub. <https://en.wikipedia.org/wiki/GitHub>. Dostopal: 11-08-2015.
- [6] How To Deploy a Rails App with Unicorn and Nginx on Ubuntu 14.04. <https://www.digitalocean.com/community/tutorials/how-to-deploy-a-rails-app-with-unicorn-and-nginx-on-ubuntu-14-04>. Dostopal: 27-08-2015.
- [7] HTML. <https://en.wikipedia.org/wiki/HTML>. Dostopal: 09-08-2015.
- [8] HTML 1.0 Tags. <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>. Dostopal: 09-08-2015.
- [9] HTML5. <https://en.wikipedia.org/wiki/HTML5>. Dostopal: 09-08-2015.

-
- [10] JavaScript. <https://en.wikipedia.org/wiki/JavaScript>. Dostopal: 09-08-2015.
 - [11] jQuery. <https://en.wikipedia.org/wiki/JQuery>. Dostopal: 11-08-2015.
 - [12] jQueryUI. <https://en.wikipedia.org/wiki/JQueryUI>. Dostopal: 11-08-2015.
 - [13] Mac OS X Leopard. https://en.wikipedia.org/wiki/Mac_OS_X_Leopard. Dostopal: 07-08-2015.
 - [14] Model-view-controller. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. Dostopal: 15-08-2015.
 - [15] MySQL Workbench. <http://www.mysql.com/products/workbench/>. Dostopal: 07-08-2015.
 - [16] MySQL Workbench. https://en.wikipedia.org/wiki/MySQL_Workbench. Dostopal: 07-08-2015.
 - [17] Nginx. <https://en.wikipedia.org/wiki/Nginx>. Dostopal: 29-08-2015.
 - [18] Object-relational mapping. https://en.wikipedia.org/wiki/Object-relational_mapping. Dostopal: 11-08-2015.
 - [19] PhpMyAdmin. <https://en.wikipedia.org/wiki/PhpMyAdmin>. Dostopal: 08-08-2015.
 - [20] Ruby Programming Language. <https://www.ruby-lang.org/en/about/>. Dostopal: 10-08-2015.
 - [21] Ruby (programming language). [https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language)). Dostopal: 10-08-2015.
 - [22] RubyMine. <https://www.jetbrains.com/ruby/>. Dostopal: 07-08-2015.

-
- [23] David Heinemeier Hansson. Ruby on Rails. https://en.wikipedia.org/wiki/Ruby_on_Rails. Dostopal: 07-08-2015.
- [24] Jesse James Garrett. Ajax: A New Approach to Web Applications. [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)). Dostopal: 09-08-2015.